Chapter: **4.0 Requirements Definition Stage**

Description: The primary goal of this stage is to develop a basis of mutual understanding between the business owner/users and the project team about the requirements for the project. The result of this understanding is an approved Requirements Specification that becomes the initial baseline for product design and a reference for determining whether the completed product performs as the system owner requested and expected. All system requirements, (e.g., software, hardware, performance, functional, infrastructure, etc.) should be included.

This stage involves analysis of the business owner/users' business processes and needs, translation of those processes and needs into formal requirements, and planning the testing activities to validate the performance of the product.

Input: The following work products provide input to this stage:

SEM Templates:

- Software Configuration Management Plan
- Maintenance Plan

PMM Templates:

- Business Case
- Concept Document (i.e., Feasibility Study)
- Project Charter (Statement of business objectives)
- Project Plan (includes Quality Management Plan)
- Security Plan

Other Inputs:

- Enterprise Architecture (EA) Solution Assessment for each potential solution option

High-Level Activities: The remainder of this chapter is divided into sections that describe specific high-level activities performed during this stage.

4.1 Requirements Management
4.2 Select Requirements Analysis Technique
4.3 Define System Requirements
4.4 Compile and Document System Requirements
4.5 Develop System Test Requirements
4.6 Develop Acceptance Test Requirements
4.7 Establish Functional Baseline

Touch Points: The following touch points are involved in the Requirements Definition Stage:

- Contracts and Procurement
    - Completion of DIT-0015a, if procuring commodities (e.g., servers, software)
    - Completion of DIT-0015b (including Statement of Work and Requirements Traceability Matrix), if procuring services (e.g., project management, application developers)
    - Utilize the services of the assigned Contract Liaison, if procuring services

- E-Michigan
    - Web review assessment by E-Michigan's webmaster to ensure ADA compliance and Michigan.gov look and feel standards. Contact E-Michigan for more information on obtaining this review.

- Enterprise Architecture (EA)
    - Use relevant EA materials (e.g., roadmaps, solution patterns) while developing Technical Requirements
    - Revise/complete EA Solution Assessment for each alternative
    - Refer to the EA TechTalk portal site

- Infrastructure Services
    - When EA Solution is complete and approved, prepare Infrastructure Services Request (ISR), and begin Hosting Solution document

- Security
    - Review MDIT and Agency security policies
    - Review State and Federal laws and regulations
    - Begin Infrastructure/Network and Data Flow Diagram

Output: Several work products are developed during this stage. The work products listed below are the minimum requirements for a large systems project. Deviations in the content and delivery of these work products are determined by the size and complexity of a project. Explanations of the work products are provided under the applicable activities described in the remainder of this chapter.

SEM Templates:

- Requirements Specification *(initial)*
- Requirements Management Checklist
- Requirements Traceability Matrix *(initial)*
- Maintenance Plan *(revised)*
- Software Configuration Management Plan *(revised)*

PMM Templates:

- Project Plan *(revised)*
- Security Plan *(revised)*

Other Outputs:

- Application Hosting document *(initial)*
- Business Continuity Plan *(revised)*
- EA Solution Assessment *(final)*
- Infrastructure Services Request (*final*)

A diagram showing the work products associated with each high-level activity is provided in *Exhibit 4.0-1, SEM Overview Diagram – Requirements Definition Stage Highlighted.*

Once the requirements are baselined, changes must be managed through the established change process, which may include a change control board.

Review Process: Quality reviews are necessary during this stage to validate the product and associated work products. The activities that are appropriate for quality reviews are identified in this chapter and Chapter 2.0, Lifecycle Model.

Structured Walkthrough (SWT)

Requirements for a peer review or a more formal structured walkthrough are documented under *Review Process* at the end of each Task, Subtask, or Activity section in this stage. The State of Michigan guide titled *Structured Walkthrough Process Guide* provides a procedure and sample forms that can be used for SWTs. This document is available on the MDIT SUITE website.

Stage Exit

Schedule a Stage Exit as the last activity of the Requirements Definition Stage to enable the project approvers to review project deliverables and provide a concur/non-concur position to the project manager. The State of Michigan guide titled *Stage Exit Process Guide* provides a procedure and sample report form that can be used for stage exits. This document is available on the MDIT SUITE website.

References: Chapter 2.0, Lifecycle Model, Q*uality Review*, provides an overview of the Quality Reviews to be conducted on a project.

Bibliography: The following materials were referenced in the preparation of this chapter.

1. Carnegie Mellon University, Software Engineering Institute, *Capability Maturity Model: Guidelines for Improving the Software Process*, Addison Wesley Longman, Inc., 1994.

2. *Software Engineering Handbook*, Chapter 4, Software Requirements Analysis.

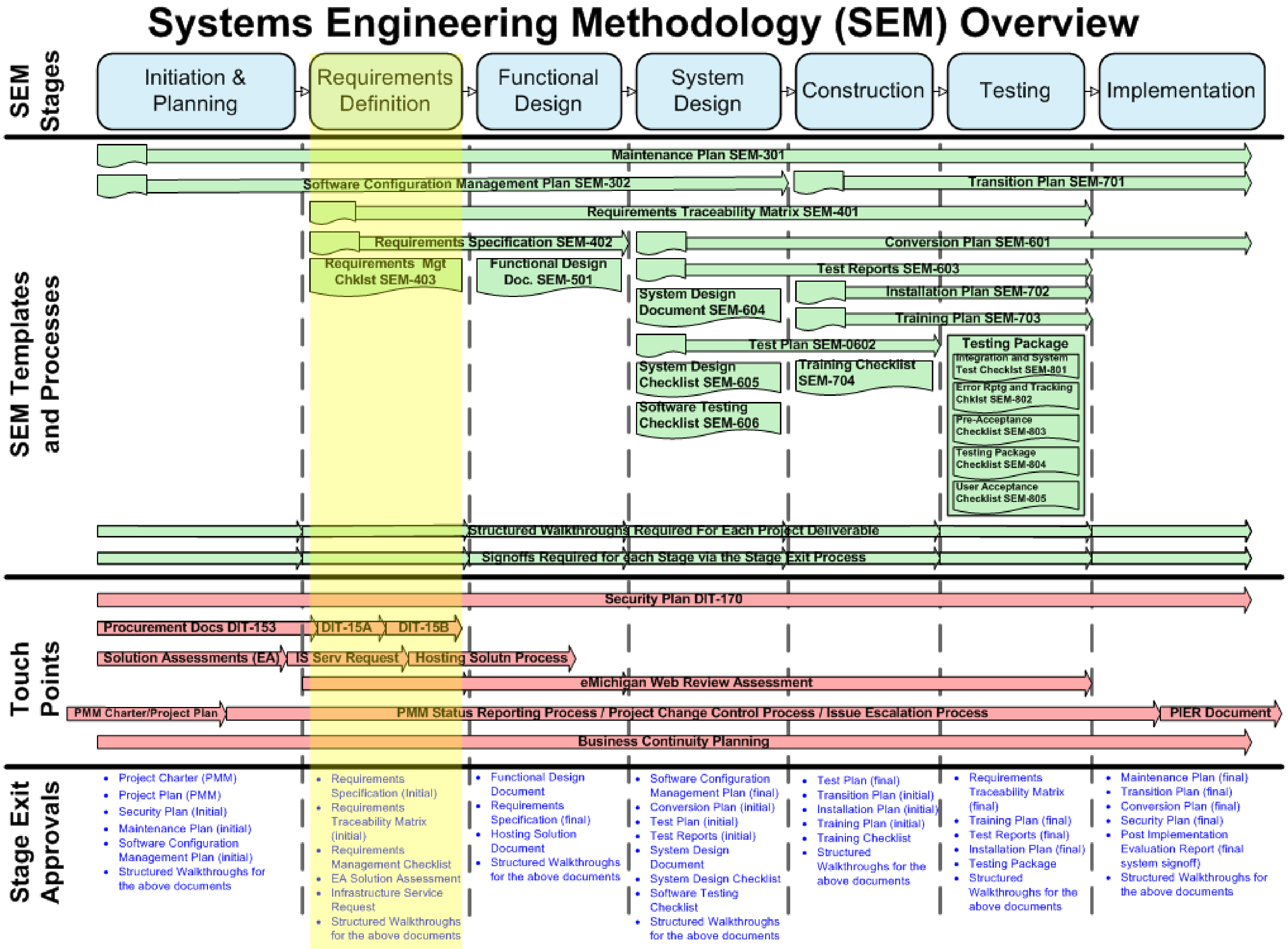
3. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Guide to Software Requirements Specifications,* ANSI/IEEE Std 830-1998, New York, 1998.

4. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standard for Developing Software Lifecycle Processes,* IEEE Std 1074-1991, New York, 1992.

5. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standard for Software Verification and Validation Plans,* ANSI/IEEE Std 1012-1998, New York, 1998.

6. U.S. Department of Energy, Nuclear Weapons Complex, Quality Managers, Software Quality Assurance Subcommittee, *Guidelines for Software Requirements Management*, July 1998.

Exhibit 4.0-1 SEM Overview Diagram – Requirements Definition Stage Highlighted



Systems Engineering Methodology (SEM) Overview

SEM Stages

Initiation & Planning → Requirements Definition → Functional Design → System Design → Construction → Testing → Implementation

SEM Templates and Processes

- Maintenance Plan SEM-301
- Software Configuration Management Plan SEM-302
- Transition Plan SEM-701
- Requirements Traceability Matrix SEM-401
- Requirements Specification SEM-402
- Conversion Plan SEM-601
- Requirements Mgt Chklst SEM-403
- Functional Design Doc. SEM-501
- Test Reports SEM-603
- System Design Document SEM-604
- Installation Plan SEM-702
- Training Plan SEM-703
- Test Plan SEM-0602
- System Design Checklist SEM-605
- Training Checklist SEM-704
- Software Testing Checklist SEM-606
- Testing Package
  - Integration and System Test Checklist SEM-801
  - Error Rptg and Tracking Chklst SEM-802
  - Pre-Acceptance Checklist SEM-803
  - Testing Package Checklist SEM-804
  - User Acceptance Checklist SEM-805
- Structured Walkthroughs Required For Each Project Deliverable
- Signoffs Required for each Stage via the Stage Exit Process

Touch Points

- Security Plan DIT-170
- Procurement Docs DIT-153 → DIT-15A → DIT-15B
- Solution Assessments (EA) → IS Serv Request → Hosting Solutn Process
- eMichigan Web Review Assessment
- PMM Charter/Project Plan → PMM Status Reporting Process / Project Change Control Process / Issue Escalation Process → PIER Document
- Business Continuity Planning

Stage Exit Approvals

Initiation & Planning:
- Project Charter (PMM)
- Project Plan (PMM)
- Security Plan (Initial)
- Maintenance Plan (initial)
- Software Configuration Management Plan (initial)
- Structured Walkthroughs for the above documents

Requirements Definition:
- Requirements Specification (Initial)
- Requirements Traceability Matrix (Initial)
- Requirements Management Checklist
- EA Solution Assessment
- Infrastructure Service Request
- Structured Walkthroughs for the above documents

Functional Design:
- Functional Design Document
- Requirements Specification (final)
- Hosting Solution Document
- Structured Walkthroughs for the above documents

System Design:
- Software Configuration Management Plan (final)
- Conversion Plan (initial)
- Test Plan (initial)
- Test Reports (initial)
- System Design Document
- System Design Checklist
- Software Testing Checklist
- Structured Walkthroughs for the above documents

Construction:
- Test Plan (final)
- Transition Plan (initial)
- Installation Plan (initial)
- Training Plan (initial)
- Training Checklist
- Structured Walkthroughs for the above documents

Testing:
- Requirements Traceability Matrix (final)
- Training Plan (final)
- Test Reports (final)
- Installation Plan (final)
- Testing Package
- Structured Walkthroughs for the above documents

Implementation:
- Maintenance Plan (final)
- Transition Plan (final)
- Conversion Plan (final)
- Security Plan (final)
- Post Implementation Evaluation Report (final system signoff)
- Structured Walkthroughs for the above documents

Activity: **4.1 Requirements Management**

Responsibility: Project Manager / Business Area Manager

Description: Requirements management is essentially a process composed of gathering, organizing, prioritizing, and documenting requirements; verifying that requirements have been captured in the product, and managing changes to requirements.

Gathering, organizing, prioritizing and documenting requirements is an interactive communication process and working relationship between stakeholders and the project team to discover, define, refine, and record a precise representation of the product requirements.

Requirements management documents the needs, expectations, and understanding of the product to be delivered and provides a framework for identifying, planning, scheduling, costing, verifying, tracing, testing, evaluating, changing, and renegotiating requirements to satisfy stakeholder needs and expectations of the project. When requirements are initially gathered, some or all will be planned for the current project (e.g., initial release). The requirements for the project are documented in the Requirements Specification document. As the project progresses, more requirements may be identified and managed through a change control process. As part of requirements management, the project manager must track requirements that are accepted for the current project and those that will be planned for subsequent releases.

Each requirement in the Requirements Specification document should be uniquely identified in a Requirements Traceability Matrix. The Requirements Traceability Matrix is a requirements management tool that ensures requirements are traced and verified through the various lifecycle stages – especially design, testing, and implementation. Requirements must be traceable from external sources such as the customer, to derived system-level requirements, to specific hardware/software product requirements.

Work Products: A substantial amount of information that is used for requirements management in later stages in the systems engineering process is gathered in the Requirements Definition Stage. The Requirements Management Checklist should be used to ensure that all requirements management activities are performed. The Requirements Traceability Matrix is a work product that is created during the Requirements Definition Stage and used to verify and validate that requirements are met and the product remains within scope. Refer to each task for information on applicable work products.

Resources:

Templates for the Requirements Traceability Matrix, Requirements Management Checklist, and Requirements Specification document are provided on the MDIT SUITE website.

Templates for the Software Change Request Form and the Software Change Control Log are provided on the MDIT SUITE website.

Carnegie Mellon University, Software Engineering Institute, *Capability Maturity Model: Guidelines for Improving the Software Process*, Addison Wesley Longman, Inc., 1994.

Tasks:

The following task is involved in requirements management.

4.1.1 Develop Requirements Traceability Matrix

Task: **4.1.1 Develop Requirements Traceability Matrix**

Description: A requirements traceability matrix is a tool used to trace project lifecycle activities and work products to the project requirements. The matrix establishes a thread that traces requirements from identification through implementation.

Every project requirement must be traceable back to a specific project objective(s) described in the Project Charter. This traceability assures that the product will meet all of the project objectives and will not include inappropriate or extraneous functionality.

All work products developed during the design, code, and testing processes in subsequent lifecycle stages must be traced back to the project requirements described in the Requirements Specification. This traceability assures that the product will satisfy all of the requirements and remain within the project scope.

It is also important to know the source of each requirement, so that the requirements can be verified as necessary, accurate, and complete. Meeting conference records, user survey responses, and business documents are typical sources for project requirements.

Work Product: Develop a matrix to trace the requirements back to the project objectives identified in the Project Charter and forward through the remainder of the project lifecycle stages. Place a copy of the matrix in the Project File. Expand the matrix in each stage to show traceability of work products to the requirements and vice versa. The requirements traceability matrix contains descriptions for each item in the matrix.

Review Process: Conduct a structured walkthrough of the Requirements Traceability Matrix to ensure that all requirements have been accurately captured.

Sample Traceability Matrix: One method for tracing requirements is a threading matrix that groups requirements by project objectives. Under each project objective, the source of the requirement, the unique requirement identification number, and the lifecycle activities are listed in columns along the top and the project requirements in rows along the left side. As the project progresses through the lifecycle stages, a reference to each requirement is entered in the cell corresponding to the appropriate lifecycle activity. A template example for the Requirements Traceability Matrix is provided on the MDIT SUITE website.

Resource: A template for the Requirements Traceability Matrix is provided on the MDIT SUITE website.

Activity: **4.2 Select Requirements Analysis Technique**

Responsibility: Project Manager/Team

Description: A requirements analysis technique is the set of data collection and analysis techniques (e.g., Joint Application Design [JAD], user interviews, and rapid prototyping) combined with the lifecycle requirements standards (e.g., tracing the requirements through all lifecycle activities) that are used to identify the project requirements and to define exactly what the product must do to meet the system owner/users' needs and expectations. When appropriate, the technique must include methods for collecting data about users at more than one geographic location and with different levels and types of needs.

The requirements analysis technique should be in harmony with the type, size, and scope of the project; the number, location, and technical expertise of the users; and the anticipated level of involvement of the users in the data collection and analysis processes. The technique should ensure that the functionality, performance expectations, and constraints of the project are accurately identified from the system owner/users' perspective. The technique should facilitate the analysis of requirements for their potential impact on existing operations and business practices, future maintenance activities, and the ability to support the system owner's long-range information resource management plans.

It is advantageous to select a technique that can be repeated for similar projects. This allows the project team and the system owner/users to become familiar and comfortable with the technique.

Discuss the analysis technique with the business owner and users to make sure they understand the process being used, their role and responsibilities in the process, and the expected format of the output (e.g., how the requirements will be organized and described).

Work Product: Create a description of the analysis technique and share it with all members of the project team, business owner, and users.

Review Process: Conduct a structured walkthrough to verify that the requirements analysis technique is appropriate for the scope and objectives of the project. A structured walkthrough is not needed when the technique has been used successfully on similar projects for the same system owner/user environment.

Activity: **4.3 Define System Requirements**

Responsibility: Project Manager/Team

Description: Use the project scope, objectives, and high-level requirements as the basis for defining the system requirements. The questions used to define the business objectives may be helpful in developing the system requirements. The goals for defining system requirements are to identify what functions are to be performed on what data, to produce what results, at what location, and for whom.

The requirements must focus on the products that are needed and the functions that are to be performed. Avoid incorporating design issues and specifications in the requirements. One of the most difficult tasks is to determine the difference between “what” is required and “how to” accomplish what is required. Generally, a requirement specifies an externally visible function or attribute of a system (i.e., “what”.) A design describes a particular instance of how that visible function or attribute can be achieved (i.e., “how to”).

Requirements should be specified as completely and thoroughly as possible. The requirements must support the business owner's business needs, information resource management long-range plans, and the enterprise (SOM-DIT) and Department-Agency missions. When requirements are being defined, it is not sufficient to state only the requirements for the problems that will be solved; all of the requirements for the project must be captured.

Attributes: Each requirement must be stated as a unique objective with the following attributes. The existence of these attributes must be verified prior to the delivery of the Requirements Specification later in the Requirements Definition Stage.

- Necessary - Absolute requirements that are to be verified are identified by "must" or "shall". Goals or intended functionality are indicated by "will".

- Correct - Each requirement is an accurate description of a feature or process of the product.

- Unambiguous - The statement of each requirement denotes only one interpretation.

- Complete - Each requirement describes one result that must be achieved by the product. The requirement should not describe the means of obtaining the result.

- Consistent - Individual requirements are not in conflict with other requirements.

- Verifiable (testable) - Each requirement is stated in concrete terms and measurable quantities. A process should exist to validate that the product (when developed) will satisfy the set of requirements.

- Modifiable - The structure and style of the requirements are such that any necessary changes to the requirements can be made easily, completely, and consistently.

- Traceable - The origin of each requirement is clear and can be tracked in future development activities and tests.

Identification System: The creation of a standard identification system for all requirements is required in order to facilitate configuration control, requirements traceability, and testing activities. The identification system must provide a unique designator for each requirement. For example, the identification system can classify the requirements by type (e.g., functional, input, or computer security). Within each type classification, the requirements can be assigned a sequential number. Select an identification system that is appropriate for the scope of the project.

Changes: As the project evolves, the requirements may change or expand to reflect modifications in the users' business plans, design considerations and constraints, advances in technology, and increased insight into user business processes. A formal change control process must be used to identify, control, track, and report proposed and approved changes. Approved changes in the requirements must be incorporated into the Requirements Specification in such a way as to provide an accurate and complete audit trail of the changes. This change control process should be an integral part of the project's Software Configuration Management Plan.

Tasks: The following tasks are involved in defining system requirements.

4.3.1 Define Functional Requirements
4.3.2 Define Input and Output Requirements
4.3.3 Define Performance Requirements
4.3.4 Define User Interface Requirements
4.3.5 Define System Interface Requirements
4.3.6 Define Communication Requirements
4.3.7 Define Computer Security and Access Requirements
4.3.8 Define Backup and Recovery Requirements
4.3.9 Define Preliminary Implementation Requirements

Task: **4.3.1 Define Functional Requirements**

Description: Functional requirements define what the product must do to support the system owner's business functions and objectives. The functional requirements should answer the following questions.

- How are inputs transformed into outputs?
- Who initiates and receives specific information?
- What information must be available for each function to be performed?

Identify requirements for all functions whether they are to be automated or manual. Describe the automated and manual inputs, processing, outputs, and conditions for all functions. Include a description of the standard data tables and data or records that will be shared with other applications. Identify the forms, reports, source documents, and inputs/outputs that the product will process or produce to help define the functional requirements.

Develop a functional model to depict each process that needs to be included. The goal of the functional model is to represent a complete top-down picture of the product.

Use flow diagrams to provide a hierarchical and sequential view of the system owner's business functions and the flow of information through the processes.

Work Product: Maintain a record of all functional requirements. Save for incorporation into the Requirements Specification.

Optional Work Product: Consider developing an optional work product that defines how the final product will operate to support the system owner organization's business functions and objectives. This user-oriented requirements manual would identify processes in a narrative form from the user's perspective and would include requirements for all functions whether they are to be automated or manual. A functional description can be developed to depict each process that will be provided. The goal is to present a complete top-down picture of the product.

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the functional requirements.

Task: **4.3.2 Define Input and Output Requirements**

Description: Describe all manual and automated input requirements for the product such as data entry from source documents and data extracts from other applications; include where the inputs are obtained.

Describe all output requirements for the product such as printed reports, display screens, and files; include who or what is to receive the output.

Data requirements identify the data elements and logical data groupings that will be stored and processed by the product. The identification and grouping of data begins during the Requirements Definition Stage and is expanded in subsequent stages as more information about the data is known.

Work Product: Maintain a record of all input and output requirements. Save for incorporation into the Requirements Specification.

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the input and output requirements.

Task: **4.3.3 Define Performance Requirements**

Description: Performance requirements define how the product must function (e.g., hours of operation, response times, and throughput under various load conditions). The information gathered in defining the project objectives can translate into very specific performance requirements; (e.g., if work performed for an organization is mission essential to the Department, the hours of operation and throughput will be critical to meeting the mission). Also, MDIT and agency policy can dictate specific availability and response times.

Work Product: Maintain a record of all performance requirements. Save for incorporation into the System Requirements Document.

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the performance requirements.

Task: **4.3.4 Define User Interface Requirements**

Description: The user interface requirements should describe how the user will access and interact with the product, and how information will flow between the user and the product.

Interface Issues: A standard set of user interface requirements may be established for the system owner organization. If not, work with the system owner and users to develop a set of user interface requirements that can be used for all automated products for the system owner's organization. A standard set of user interface requirements will simplify the design and code processes, and ensure that all automated products have a similar look and feel to the users. When other constraints (such as a required interface with another application) do not permit the use of existing user interface standards, an attempt should be made to keep the user interface requirements as close as possible to the existing standard.

The following are some of the issues that should be considered when trying to identify user interface requirements.

- The users' requirements for screen elements, navigation, and help information.
- The standards issued by the SOM, MDIT and agencies that apply to user interfaces.
- The classification of the users who will access and use the product.
- The range of work that the users will be performing with the product.

Define the user interface requirements by identifying and understanding what is most important to the user, not what is most convenient for the project team.

Work Product: Maintain a record of all user interface requirements. Save for incorporation into the Requirements Specification.

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the user interface requirements.

Task: **4.3.5 Define System Interface Requirements**

Description: The hardware and software interface requirements must specify hardware and software interfaces required to support the development, operation, and maintenance of the product.

The following information should be considered when defining the hardware and software interface requirements.

- Business owner's and users' IT environment.
- Existing or planned software that will provide data to or accept data from the product.
- Other organizations or users having or needing access to the product.
- Purpose or mission of interfacing software.
- Common users, data elements, reports, and sources for forms/events/outputs.
- Timing considerations that will influence sharing of data, direction of data exchange, and security constraints.
- Development constraints such as the operating system, database management system, language compiler, tools, utilities, and network protocol drivers.
- Standardized system architecture defined by hardware and software configurations for the affected organizations, programmatic offices, sites, or telecommunications operations.

Work Product: Maintain a record of all system interface requirements. Save for incorporation into the Requirements Specification.

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the system interface requirements.

Task: **4.3.6 Define Communication Requirements**

Description: The communication requirements define connectivity and access requirements within and between user locations and between other groups and applications.

The following factors should be considered when defining communication requirements.

- Communication needs of the user and customer organizations.
- User organization's existing and planned communications environment (e.g., telecommunications; LANs, WANs, and dial-up).
- Projected changes to the current communication architecture, such as the connection of additional local and remote sites.
- Limitations placed on communications by existing hardware and software including:
  - user systems
  - applications that will interface with the product
  - organizations that will interface with the product
- SOM, MDIT and agency standards that define communication requirements and limitations.
- Future changes that may occur during the project.

Work Product: Maintain a record of all communication requirements. Save for incorporation into the Requirements Specification.

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability accuracy, and completeness of the communications requirements.

Task: **4.3.7 Define Computer Security and Access Requirements**

Description: Develop the computer security requirements in conjunction with the system owner's Office of Enterprise Security Liaison and Agency Security Office, and other stakeholders who provide competent input in the information system security area. This involvement affords early determination of classifications and levels of access protection required for the product.

If a product under development processes sensitive personal information, appropriate safeguards must be established to protect the information from accidental disclosure. Refer to the Department of Management and Budget website for guidance on the Privacy Act.

Implement applicable security procedures to assure data integrity and protection from unauthorized disclosure, particularly during development efforts. The organization that owns the data defines the data classification. The project team must be aware of all the types of data and of any classified or proprietary algorithms used in the product.

Procedure: Use the following procedure to determine computer security requirements.

1. Identify the types of data that will be processed by the product.
2. Determine preliminary data protection requirements.
3. Coordinate with the owner of the host platform to identify existing supporting computer security controls, if applicable.
4. Incorporate security requirements into the Requirements Specification.

Sample Access Control Questions: The following list provides sample questions that can be used to help define the access controls for the product.

- What access restrictions are placed on the users by their organization or programmatic office?
- What are the audit and other checking needs for the product?
- What separation of duties, supervisory functions related to control, operating environment requirements, or other functions will impact the product?
- What measures will be used to monitor and maintain the integrity of the product and the data from the user's viewpoint?

Work Product: Maintain a record of all security and access requirements. Save for incorporation into the Requirements Specification.

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the computer security and access requirements.

Resources:

- MDIT Office of Enterprise Security Resource Guide
- MDIT Office of Enterprise Security website

Task: **4.3.8 Define Backup and Recovery Requirements**

Description: Review any applicable existing Disaster Recovery agreements, Continuity of Operations agreements, and Service Level Agreements. Amend if needed to incorporate specific requirements for data backup, recovery and operational status for the product. Ensure that any mission essential systems are included in the Business Continuity Plan.

Work Product: Maintain a record of all data backup, recovery and operational start up requirements. If a product is determined to be mission essential, a Business Continuity Plan must be developed. If the product is not mission essential, a Continuity of Operations Statement within the plan is required. Place a copy of the Business Continuity Plan in the Project File.

Review Process: Conduct structured walkthroughs as needed to assure the necessity, testability, accuracy, and completeness of the backup and recovery requirements.

Resource: A template for the Business Continuity Plan is available on the DMB website.

Task: **4.3.9 Define Preliminary Implementation Requirements**

Description: Describe the requirements anticipated for implementing the product (e.g., user production cycle). The high-level implementation requirements are identified early in the lifecycle to support decisions that need to be made for the information systems engineering approach. The implementation requirements are expanded into a full implementation approach during the design stages.

The following paragraphs provide highlights of some of the implementation requirements that need to be considered.

Operating Environment: Identify any capacity restrictions on the existing hardware or software that needs to be addressed and identify any hardware or software that needs to be acquired (e.g., communication hardware, file servers, commercial off-the-shelf software, network interface cards, and LAN utilities).

Acquisition: If hardware or software must be acquired, identify the necessary acquisition activities. These activities include preparing specifications, estimating costs, scheduling procurement activities, selection, installation, and testing.

Conversion: Identify requirements for converting data from an existing or external application to the new product. Consider requirements for data entry, data protection, computer time, conversion programs, personnel, and other resources that will be needed. Also identify the requirements for the conversion of software, if necessary. Implementing a new application may involve converting software from one environment to another or modifying software to interface with other applications. Include requirements for testing the conversion process and validating that it was successfully accomplished.

Installation: Identify the installation requirements for any new hardware, operating system, or software. For hardware installations, consider environmental factors such as air conditioning, power supply, and security requirements. For software installations, consider proprietary software such as database management systems. For application software, consider the installation of the application's programs, parallel operation of the old and new applications, or the cutover from a test to a production environment. Hardware and software installation must be coordinated with the work cycles of the user organization to create a minimum of disruption, and to assure that data are available as needed. Installation must be scheduled to assure that, when data conversion is necessary, the needed data are protected.

Training: Identify the specific training needs for various categories of users and administrators. Also identify training requirements for personnel (e.g., agency staff, Michigan businesses, and Michigan citizens) time, computer time, training facilities, and training database(s).

Documentation: Identify requirements for the development and distribution of operational documentation for support personnel and user documentation. Operational documentation may include job control procedures and listings, operational instructions, system administration responsibilities, archiving procedures, and error recovery. User documentation includes the users manual, step-by-step instructions, online documentation, and online help facilities.

Work Product: Maintain a record of all implementation requirements. Save for incorporation into the Requirements Specification.

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the implementation requirements.

Activity: **4.4 Compile and Document System Requirements**

Responsibility: Project Manager/Team

Description: Compile the requirements gathered during the requirements analysis process in preparation for the development and delivery of the draft Requirements Specification. The following steps should be performed as part of the requirements compilation activity.

- Select and use a standard format for describing the requirements. Ensure compliance with Enterprise Architecture and any site specific standards.

- Present the logical and physical requirements without dictating a physical design or technical solutions.

- Write the requirements in non-technical language that can be fully understood by the system owner and users.

- Organize the requirements into meaningful groupings (e.g., all security-related requirements or all requirements for generating reports).

- Develop a numbering scheme for the unique identification of each requirement.

- Select a method for: (1) tracing the requirements back to the sources of information used in deriving the requirements (e.g., specific system owner/user project objectives); and (2) threading requirements through all subsequent lifecycle activities (e.g., testing).

Activity: **4.5 Develop System Test Requirements**

Responsibility: Project Manager

Description: The System Test Requirements section of the Requirements Specification document is a narrative and tabular description of the test activities planned for the project during development or enhancement. System Test Requirements should establish the testing necessary to validate that the project requirements have been met and that the deliverables are at an acceptable level in accordance with existing standards. System Test Requirements also ensure that a systematic approach to testing is established and that the testing is adequate to verify the functionality of the product.

System Test Requirements includes the resources, project team responsibilities, and management techniques needed to plan, develop, and implement the testing activities that will occur throughout the lifecycle. If individuals outside of the project team perform system and acceptance testing, the plan includes the responsibilities and relationships of external test groups.

In this stage, System Test Requirements are written at a high level and focus on identifying test techniques and test phases. Detailed information about test products (i.e., test plans, test procedures, and test reports) is added to the System Test Requirements as the project progresses through subsequent lifecycle stages.

Development of the System Test Requirements is the responsibility of the project manager. If a test group outside the project team will be involved in any test phase, the project manager must coordinate the System Test Requirements with each test group.

The System Test Requirements must be reviewed and approved by the system owner prior to conducting any tests.

Notes: For small projects, formal System Test Requirements may not be necessary; however, a test approach and testing are required and must be documented.

Typically, System Test Requirements cover all test phases including integration, system, and acceptance.

Work Product: When the System Test Requirements are completed, they should contain the following information:

- Describe the occurrence and timing of the test phases in the lifecycle and the entrance and exit criteria for each test phase.

- Specify the test products at each test phase. Describe the types and scope of the testing activities to be performed on each component of the application and the group who is responsible to produce them.

- Map what requirements are verified in what test phase.

- Establish the criteria for evaluating the test results of each test phase.

- Make an initial determination of the resources necessary to accomplish the testing.

- Identify the appropriate person or group to conduct each type of testing activity.

- Outline the test environment (hardware, software, test tools, and data) needed to conduct the tests.

- Develop a preliminary schedule for executing the test activities.

Review Process: Conduct structured walkthroughs to assure the System Test Requirements adequately describe all testing activities, test schedules, test products, test responsibilities, the testing methodology, and the required resources.

Resources: The Requirements Specification template includes a section for System Test Requirements. The Requirements Specification Template is available on the MDIT SUITE website.

Tasks: Preparation of the System Test Requirements involves the following tasks.

4.5.1 Identify Test Techniques
4.5.2 Identify Test Phases
4.5.3 Identify Test Environment Requirements

Task: **4.5.1 Identify Test Techniques**

Description: The System Test Requirements should specify the testing techniques planned for the project including the types of tests required, test documents, test methods, and test data collection. Each test from integration through acceptance testing is specified in terms of entrance and exit criteria and the expected level of involvement from the project team, test group, and other functional areas.

Integration tests with appropriate data must be developed to exercise and validate all specified application requirements, functions, and objectives. System and acceptance tests validate that the integrated system meets the requirements.

Each type of test must use controlled computer generated or live data as specified. The test data must be prepared to include values that will verify the functional capabilities of the test component, identify its limitations and deficiencies (if any), exercise its capabilities, and verify that the component performs its intended function as required.

If pilot testing or a phased implementation is required for the product, the System Test Requirements should include such requirements. In the case of an implementation involving phased releases, the plan should include the requirements for regression testing of the complete application as new elements are introduced.

For each type of test conducted, the test results are compared with the expected results. Discrepancies are identified and any problems resolved. Retesting is required to verify that the problem solution eliminates the problem and does not introduce new errors. The final test results are accompanied by a completed test results/error log form. This form is completed by the individual(s) responsible for testing and attached to the documents that certify the completion of each type of test.

Task: **4.5.2 Identify Test Phases**

Description: The product should be tested in sequential phases: integration, system, and acceptance. Some projects may require additional types of tests. The test phases are described below.

Integration Test Phase: Integration testing is an orderly progression of testing in which software elements, hardware elements, or both are combined and tested to evaluate the interaction between them. Each program/module must be tested. Integration testing is required to validate that groups of related programs, when combined to establish an integrated functional module of code, interface properly, and perform the functions for which they were designed. Examine the source program/module statements to ensure that the program logic meets the requirements of the design and that the application satisfies an explicit functional requirement. This test phase is performed by the project team.

System Test Phase: The system test phase tests the integrated hardware and software to verify that the product meets its specified requirements and operates successfully on the host platform. This test phase is required to validate, when the entire product is loaded onto the host platform, that the proper initialization is performed; decision branching paths are appropriate; and all functions are performed as specified in the Requirements Specification. System testing validates that the product produces the required outputs and interfaces properly with other systems with which the product gives or receives data; that transaction response times meet user expectations; and machine resource allocation and utilization are within expected norms. This test phase can be performed by the project team or by an independent test group with support from the project team.

Acceptance Test Phase: Acceptance testing is conducted to determine whether a product satisfies its acceptance criteria and to enable the system owner's organization to determine whether to accept the product. The acceptance test is required to validate that the system, its related documentation and tools, satisfy all of the specified requirements and objectives of the system owner's organization, State of Michigan standards, the Requirements Specification, and the design criteria. Acceptance testing will include tests of all intrasystem interfaces; and the use of all manuals, documentation, procedures, and controls. This test phase can be performed by the project team with system owner and user observers or by system owner and user representatives with support from the project team.

Task: **4.5.3 Identify Test Environment Requirements**

Description: The System Test Requirements should outline what is needed to perform testing activities throughout the project lifecycle including personnel, hardware, software, space, and other environmental requirements. As much testing as possible should be performed on the same equipment that will be used for the production system. In many cases, this information is not fully known until the System Design Stage.

The following are some of the considerations for test environment requirements.

- Evaluate automated testing tools for the following:
  - Generation of test scripts
  - Creation of result and error repositories
  - Consideration of each tool's benefits and costs
  - Use of simulators
- Determine local area network, wide area network, and metropolitan area network testing environment(s), as needed
- Determine test lab, data generation, and error correction support
- Identify Beta test sites

Activity: **4.6 Develop Acceptance Test Requirements**

Responsibility: Project Team

Description: The Acceptance Test Requirements section of Requirements Specification document is a description of the test activities planned for project acceptance. The Acceptance Test Requirements should establish the testing necessary to validate that the project requirements have been met and that the deliverables are at an acceptable level in accordance with existing standards. The Acceptance Test Requirements also assures that a systematic approach to acceptance testing is established and that the testing is adequate to verify the functionality of the product.

The complete set of system requirements and acceptance criteria form the basis for determining the overall approach to acceptance testing and the specific testing and examination methods. Features of the installation site and the system affect how the acceptance testing will be done. Unique arrangements may be necessary when the product cannot be completely installed and executed in a live environment. Multiple configurations may have to be distributed at several installation sites.

When a new system is a replacement for one already in use, the acceptance test must assure the integrity of the users' business operations while placing the replacement into operation. For example, the old system and the new system are used in parallel until complete functionality has been verified. In some cases, the acceptance process may take several months to assure that a complete business or accounting cycle has occurred. This concern will influence the approach to acceptance testing.

Work Product: Acceptance testing must be documented carefully with traceability of test cases to the requirements and acceptance criteria established by the system owner. At a minimum, the Acceptance Test Requirements should address the following requirements.

- Identification of the personnel involved in the acceptance test process and their testing responsibilities. If individuals outside of the project team perform acceptance testing, include the responsibilities and relationships of external test groups.

- Traceability of test designs and cases to requirements.

- The objectives and constraints for each test.

- Complete test cases and test procedures including inputs and expected outputs for each test case.

- Descriptions of error reporting, analysis, and resolution.
- Location(s) where testing will occur, the testing approach, type of facilities, and tester training.
- Acquisition of special purpose testing equipment, tools, and software.
- Resources and cost estimation to accomplish testing.

Review Process: Conduct structured walkthroughs to assure the draft Acceptance Test Requirements adequately describes all testing activities, test schedules, test products, test responsibilities, the testing methodology, and the required resources.

Resources: The Requirements Specification template includes a section for Acceptance Test Requirements. The Requirements Specification Template is available on the MDIT SUITE website.

Activity: **4.7 Establish Functional Baseline**

Responsibility: Project Manager/Team

Description: The functional baseline, sometimes called a system requirements baseline, is the main technical work product of the Requirements Definition Stage. The system requirements are baselined after the system owner's formal approval of the Requirements Specification. Once the requirements are baselined, any changes to the requirements must be managed under change control procedures established in the Software Configuration Management Plan. Approved changes must be incorporated into the Requirements Specification.

Work Product: Prepare the final Requirements Specification and submit to the system owner and users for their review and approval. The approved Requirements Specification is the official agreement and authorization to use the requirements for the product design. Approval implies that the requirements are understood, complete, accurate, and ready to be used as the basis for the subsequent lifecycle stages.

It is important for the system owner/users to understand that changes to the approved Requirements Specification affect the project scope and therefore can change the project cost, resources, or schedule. It is the responsibility of the project manager and project team to identify system owner/user requested changes that would result in a change of project scope; evaluate the potential impact to the project costs, resources, or schedule; and notify the system owner of the project planning revisions that will be required to accommodate their change requests. The Requirements Specifications are validated using the Requirements Management Checklist.

Place a copy of the approved Requirements Specification and the Requirements Management Checklist in the Project File.

Review Process: The Requirements Specification should be reviewed by the system owner and users. After making the changes needed to resolve problems found during the review, the functional baseline is formally established upon receipt of the system owner's approval.